Modeling and Control of Wastewater Treatment Process

by

Adi Hasif bin Talib (9780)

Dissertation submitted in partial fulfilment of
the requirements for the
Bachelor of Engineering (Hons)
(Chemical Engineering)

MAY 2011

Universiti Teknologi PETRONAS

Bandar Seri Iskandar

31750 Tronoh

Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Modeling and Control of Activated Sludge Process

by

Adi Hasif bin Talib

A project dissertation submitted to the
Chemical Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(CHEMICAL ENGINEERING)

Approved by,

_____

(Dr. Nooryusmiza Yusoff)

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

May 2011

ABSTRACT

Studies and research in modeling and control tools of the biological wastewater treatment process is very significant to improve the conventional operation control strategies and eventually giving big positive impact to our own environment. Hence, the objectives of this work are:

i. To study wastewater treatment process
ii. To model and simulate an activated sludge process
iii. To apply control strategies on the model

A wastewater treatment plant has been studied and activated sludge process has become the subject of concern as it is the most widely used technology for biological treatment in a wastewater treatment plant. The activated sludge plant has been modeled following the Benchmark Simulation Model No. 1 (BSM1) developed by COST 682 Working Group No. 2 to provide standard process assessment and besides, this model is accepted internationally. The model is implemented using MATLAB/Simulink Software to capture the mathematical model describing the process. The control of dissolved oxygen level in the reactors plays an important role in the operation, thus conventional PI control strategy had been applied to the model to control the dissolve oxygen (DO) level in the aerated reactor. To further improvement of the DO control in the system, a new approach of advance control strategy had been also applied to the system. Model Predictive Control (MPC) is introduced to the activated sludge plant. The control strategies had been evaluated and comparison had been made between the conventional PI controller and the MPC. The results showed that the system shows control improvement with the MPC implemented to the plant.

Research study and literature review on the topic will be discussed in Chapter 2 of the report. All resources had been used wisely to obtain the most information regarding wastewater treatment process and its control strategies. Methodology of this project is represented in Chapter 3 of this report. Gantt chart is also included to present the activity conducted throughout the semester which in all, most of them have been executed successfully. The results of the project are shown in Chapter 4. Finally in Chapter 5, conclusion of the project is represented with recommendation for future work.

ACKNOWLEDGEMENT

Foremost gratitude goes to my supervisor, Dr Nooryusmiza Yusoff for his great supervision and guidance towards me to work on Final Year Project 1 (FYP1). His tolerance throughout is very much appreciated which I would not have been able to repay them.

I would also like to extend my heartfelt appreciation to Dr. Mohanad El-Harbawi and Dr. Lukman Ismail for giving so much information to help me in the process of this project.

Besides that, my special gratitude is also extended to all my friends and family whom has been giving strong support and motivation throughout the completion of the project.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background Study

Water pollution has been a topic of concern now and most probably to more years to come in the future. The increase public awareness of the seriousness of this problem has raised the bar in more stringent effluent regulations. This has considerably increased the requirements imposed on treatment plants. To comply with the challenge as well as to contain plant cost, activated sludge process has always been opted among others to accomplish water treatment economically with the process of biological oxidation. Activated sludge technology has been used since the 20th century as it is cost-effective, it can be adapted to any kind of wastewater, reliable and has the capacity of producing high quality effluent. For all these reasons, our interest during this work will focus on studying and implementing further improvements to this technology. In the activated sludge process, a bacterial biomass suspension (the activated sludge) is responsible for the removal of pollutant.

Figure 1.1 below is showing the basic layout of the considered wastewater treatment technology, the activated sludge process.

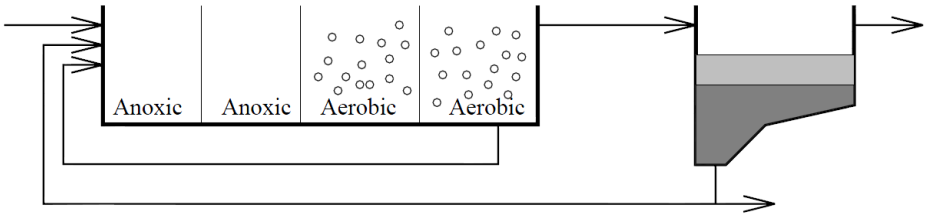

Figure 1.1: Basic layout of the activated sludge process.

The activated sludge is using microorganism to oxidize and mineralize organic component and nitrogen. Its biological process of nitrification-denitrification allows removal of high nitrogen load in wastewater plant. First the ammonium presence in the influent is oxidized forming nitrate in the aerated tank (nitrification). Then the bacteria respire with nitrate into nitrogen gas in the anoxic tank (denitrification).

The oxygen presence in the aerated zone is very important and has significant influence in controlling behaviour and activity of the nitrogen removal process. Sufficient amount of oxygen has to be supplied to the microorganism in the sludge for degradation of the organic matter. However, an excessive supply of oxygen may deteriorate the sludge quality, reduce the nitrification efficiency as well as imposing high cost to the plant for the high airflow rate. Hence, it is of interest to control the oxygen supply to the system.

1.2 Problem Statement

The application of process control over a wastewater treatment plant has been encountering difficulties in the past years such as:

i. Studies have shown that even well attended plant fails to meet the requirement of quality standard.
ii. More stringent effluent regulation has been implemented.
iii. Difficulties in plant modeling to simulate the actual plant due to its vast process parameters and complexity of dynamics.
iv. Difficulties in controlling the process due to the numerous parameters influencing the process (for instance influent flow rate, organic and nutrient load variations, toxicity, temperature).

1.3 Objective and Scope of Study

The objectives of this research are:

i. To study wastewater treatment process
ii. To model and simulate an activated sludge process
iii. To apply control strategies on the model

The whole project will start on wastewater treatment process analysis and the study about modeling parameters and control tools is focused in the biological treatment area which is the activated sludge process.

CHAPTER 2

LITERATURE REVIEW

2.1 Modeling Activated Sludge Process

As illustrated in the general layout of an activated sludge process, the system composes of two main units which are the biological reactors and a settler. To describe this process, benchmark simulation process has been developed to assist with the complexity of the process. To keep the simplicity of this research, the Benchmark Simulation Model 1 has been selected to direct the modeling simulation process to an accepted standard.

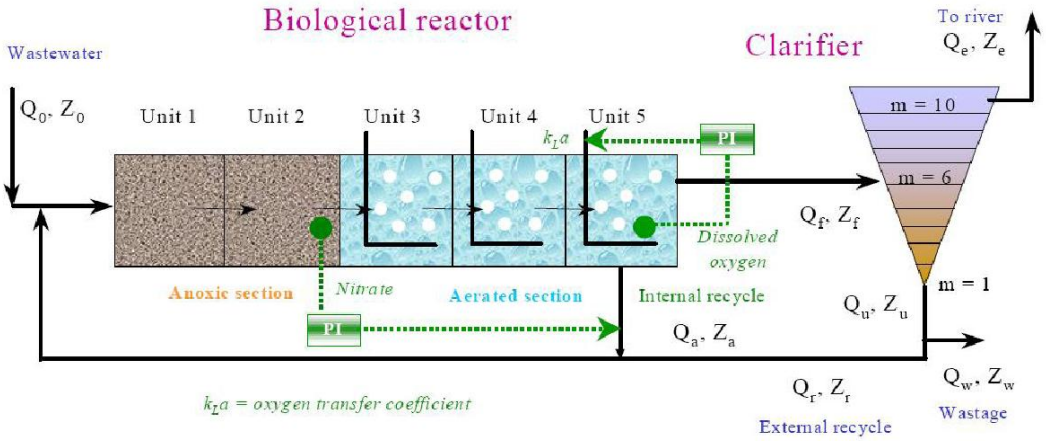

Figure 2.1: Benchmark Simulation Model 1 (BSM1) layout.

The plant consists of a five-compartment activated sludge reactor with two anoxic tanks followed by three aerobic tanks. The plant thus combines nitrification with predenitrification in a configuration that is commonly used for achieving biological nitrogen removal in full-scale plants. The activated sludge reactor is then followed by a secondary clarifier or a settler to separate the sludge and the treated water. Two control strategies are implemented in this structure. The first controller is to control the dissolve oxygen level by manipulating the oxygen transfer coefficient and the second controller serves the purpose of controlling the nitrate level in the last anoxic tank by manipulation of the internal recycle flow rate. (Copp, 2002)

There are several models described by International Water Association (IWA) of the biological process in the activated sludge plant. The models of Activated Sludge Model (ASM) family (ASM1, ASM2, ASM2d, ASM3) are used in most of the modeling and simulation studies; additionally, those are considered state-of-art models of activated sludge processes and are used in most of the commercial simulation platforms:

i. Activated Sludge Model No. 1 (ASM1)
    - The aim of Activated Sludge Model 1 was to create a common platform that could be used for future development of models for nitrogen removal activated sludge processes; furthermore, the aim was to develop a model with a minimum of complexity. In ASM1 two kinds of substrate which are readily and slowly biodegradable COD (RBCOD and SBCOD) are introduced and a hydrolysis process is included. In ASM1 it is also assumed that slowly biodegradable substrate consists fully of particulate substrate (X_S). ASM1 includes nitrogen and organic matter removal with simultaneous consumption of oxygen and nitrate as electron acceptors; however, it does not contain biological phosphorous removal. ASM1 was developed mainly for municipal activated sludge plants. (Mulas, 2006)
ii. Activated Sludge Model No. 2 (ASM2)
    - Activated Sludge Model No. 2 (ASM2) was published in 1995, which include both nitrogen removal and biological phosphorus removal. The role of denitrification in relation to biological phosphorus removal was still unclear, and hence the element is not included. However, the development in research was fast, and denitrifying PAOs (phosphorus accumulating organisms) were needed for simulation of many results from research and practice. Because of this, the ASM2 model was expanded in 1999 into the ASM2d model, where denitrifying PAOs were included. (Mulas, 2006)
iii. Activated Sludge Model No. 3 (ASM3)
    - Activated Sludge Model No. 3 (ASM3) describes the same processes as ASM1; however, ASM3 was introduced to correct the deficiencies of ASM1. The most important reason for introducing ASM3 was the

recognition of importance of three rates of oxygen consumption in the process: the rapid rate of oxygen consumption for degradation of RBCOD, slow rate associated with degradation of SBCOD, and even slower endogenous oxygen uptake rate (OUR). In ASM1 there is only one oxygen consuming process, which makes calibration of the model very difficult. Calibration of ASM3 should be easier mainly because of converting the circular growth death growth (death regeneration) model by endogenous respiration model. (Mulas, 2006)

In this research, ASM1 model is adopted to describe the biological phenomena taking place in the biological reactor as it is widely use and has been largely accepted by the international community. (Henze, Grady, Jr., Gujer, Marais, & Matsuo, 1987)

2.2 State Variables

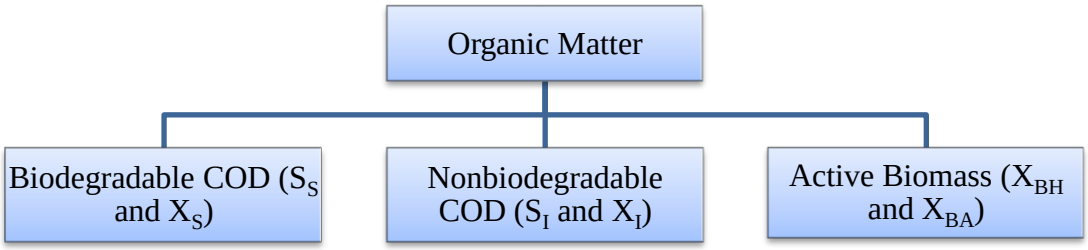

Figure 2.2: Components of organic matter in ASM1.

Table 2.1: ASM1 State Variables.

State Variables	Notation	Unit
Soluble inert organic matter	S_I	$gCOD/m^3$
Readily biodegradable substrate	S_S	$gCOD/m^3$
Particulate inert organic matter	X_I	$gCOD/m^3$
Slowly biodegradable substrate	X_S	$gCOD/m^3$
Active heterotrophic biomass	X_{BH}	$gCOD/m^3$
Active autotrophic biomass	X_{BA}	$gCOD/m^3$
Particulate product from biomass decay	X_P	$gCOD/m^3$
Dissolved Oxygen	S_O	gO_2/m^3
Nitrate and Nitrate Nitrogen	S_{NO}	gN/m^3
Free and Ionized Ammonia	S_{NH}	gN/m^3
Soluble biodegradable organic Nitrogen	S_{ND}	gN/m^3
Particulate biodegradable organic Nitrogen	X_{ND}	gN/m^3
Alkalinity	S_{ALK}	Molar units

Apart from these variables to describe the process, there are also composite variables to combine them into forms that are measured in reality such as Chemical Oxygen Demand (COD), Total Suspended Solids (TSS), and Total Nitrogen (TN).

$$COD = S_I + S_S + X_I + X_S + X_{BH} + X_{BA} + X_P \quad [\text{gCOD/m}^3]$$
$$TSS = 0.75(X_I + X_S + X_P) + 0.9(X_{BH} + X_{BA}) \quad [\text{gSS/m}^3]$$
$$TN = S_{NO} + S_{NH} + S_{ND} + X_{ND} + i_{XB}(X_{BH} + X_{BA}) + i_{XP}(X_P + X_I) \quad [\text{gN/m}^3]$$

2.3 Biological Parameter Values

Corresponding to the temperature of 15^{o}C, the stoichiometric and kinetic parameters are listed in Table 4.2 and Table 4.3 below.

Table 2.2: Stoichiometric Parameters.

Parameter	Unit	Value
Y_A	g cell COD formed (g N oxidized)$^{-1}$	0.24
Y_H	g cell COD formed (g COD oxidized)$^{-1}$	0.67
f_P	dimensionless	0.08
i_{XB}	g N. (g COD)$^{-1}$ in biomass	0.08
i_{XP}	g N. (g COD)$^{-1}$ in particulate products	0.06

Table 2.3: Kinetic Parameters.

Parameter	Unit	Value
μ_H	d^{-1}	4.0
K_S	g COD.m^{-3}	10.0
$K_{O,H}$	g (-COD).m^{-3}	0.2
K_{NO}	g NO_3-N.m^{-3}	0.5
b_H	d^{-1}	0.3
η_g	dimensionless	0.8
η_h	dimensionless	0.8
k_h	g slowly biodegradable COD.(g cell COD.d)$^{-1}$	3.0
K_X	g slowly biodegradable COD.(g cell COD)$^{-1}$	0.1
μ_A	d^{-1}	0.5
K_{NH}	g NH_3-N.m^{-3}	1.0
b_A	d^{-1}	0.05
$K_{O,A}$	g (-COD).m^{-3}	0.4
k_a	m^3 (g COD.d)$^{-1}$	0.05

2.4 Dynamic Model and Processes

2.4.1 Growth of Biomass

Aerobic growth of heterotrophic biomass

The degradation of soluble readily biodegradable substrate under consumption of oxygen leads to growth of heterotrophic biomass. Ammonia is used as nitrogen source and incorporated into cell mass. The process is generally the main contributor to the production of biomass and removal of COD. The process is described with the monod function (Copp, 1998):

$$\rho_1 = \mu_H \left(\frac{S_S}{K_S + S_S}\right)\left(\frac{S_O}{K_{OH} + S_O}\right) X_{BH}$$

Anoxic growth of heterotrophic biomass

In the absence of oxygen the heterotrophic organisms use SS as substrate and nitrate as their electron acceptor. The result is growth of biomass and production of nitrogen gas due to denitrification. The process is also described with the monod function and the kinetic rate expression is multiplied by a factor η_g (<1) (Copp, 1998):

$$\rho_2 = \mu_H \left(\frac{S_S}{K_S + S_S}\right)\left(\frac{S_O}{K_{OH} + S_O}\right)\left(\frac{S_{NO}}{K_{NO} + S_{NO}}\right)\eta_g X_{BH}$$

Aerobic growth of autotrophic biomass

Ammonia nitrogen is oxidised to nitrate via nitrification. This results in growth of autotrophic biomass and a high oxygen demand. Ammonia is also the nitrogen source for synthesis and incorporated into the cell mass. This process as well is described with a monod function (COST, 1998):

$$\rho_3 = \mu_A \left(\frac{S_{NH}}{K_{NH} + S_{NH}}\right)\left(\frac{S_O}{K_{OA} + S_O}\right) X_{BA}$$

2.4.2 Decay of Biomass

Decay of heterotrophic biomass

The organisms are assumed to die at a certain rate and part of the decay results in release of slowly biodegradable substrate. The rest is considered non-biodegradable and adds to the XP fraction. The process is assumed to have the same rate during aerobic, anoxic and anaerobic conditions and is described by:

$$\rho_4 = b_H X_{BH}$$

Decay of autotrophic biomass

This process described in a similar was as the decay heterotrophs

$$\rho_5 = b_A X_{BA}$$

2.4.3 Ammonification of soluble organic nitrogen

Biodegradable soluble organic nitrogen is converted to free and saline ammonia in a first order process (COST, 1998):

$$\rho_6 = k_a S_{ND} X_{BH}$$

2.4.4 Hydrolyses

Hydrolysis of entrapped organic matter

Slowly biodegradable substrate entrapped in the sludge breaks down extracellulary. This process produces readily biodegradable substrate that is available for the organisms for growth. It occurs both under aerobic and anoxic conditions and is modelled on surface reaction kinetics. Under anoxic conditions the rate is reduced with a factor η_h (<1) (COST, 1998):

$$\rho_7 = k_h \frac{X_S/X_{BH}}{K_X + (X_S/X_{BH})} \left[\left(\frac{S_O}{K_{OH} + S_O} \right) + \eta_h \left(\frac{K_{OH}}{K_{OH} + S_O} \right) \left(\frac{S_{NO}}{K_{NO} + S_{NO}} \right) \right] X_{BH}$$

Hydrolysis of entrapped organic nitrogen

This process of biodegradable particulate organic nitrogen breaking down to soluble organic nitrogen is described as the rate for the entrapped organics (COST, 1998):

$$\rho_8 = k_h \frac{X_S/X_{BH}}{K_X + (X_S/X_{BH})}\left[\left(\frac{S_O}{K_{OH} + S_O}\right) + \eta_h \left(\frac{K_{OH}}{K_{OH} + S_O}\right)\left(\frac{S_{NO}}{K_{NO} + S_{NO}}\right)\right] X_{BH} \frac{X_{ND}}{X_S}$$

2.5 Model Formulation

The final description of the model is based on the dynamic processes and builds a set of ordinary differential equations. The first one represents the behaviour of the heterotrophic biomass concentrations and shows how it is affected by the three processes; aerobic growth, anoxic growth and decay (Jeppson, 1996):

$$\frac{dX_{BH}}{dt} = \rho_1 + \rho_2 - \rho_4$$

The second one describes the autotrophic biomass concentration and is slightly simpler because the autotrophs do not grow in anoxic environment.

$$\frac{dX_{BA}}{dt} = \rho_3 - \rho_5$$

The third differential equation describes how the readily biodegradable substrate behaves. The concentration is reduced by growth of heterotrophic bacteria and increased by hydrolysis of slowly biodegradable substrate.

$$\frac{dS_S}{dt} = \rho_7 - (\rho_1 + \rho_2)$$

The next differential equation is describing how the slowly biodegradable substrate is increasing by the recycling of dead bacteria and decreasing by the hydrolysis process.

$$\frac{dX_S}{dt} = (1 - f_P)(\rho_4 + \rho_5) - \rho_7$$

The fifth model equation describes how the concentration of inert particulate products is arising from biomass decay.

$$\frac{dX_P}{dt} = f_P(\rho_4 + \rho_5)$$

The sixth equation is describing how the concentration of particulate organic nitrogen behaves similar to the slowly biodegradable substrate.

$$\frac{dX_{ND}}{dt} = (i_{XB} - f_P i_{XP})(\rho_4 + \rho_5) - \rho_7$$

Soluble organic nitrogen is affected by ammonification and hydrolysis which shows in the seventh equation.

$$\frac{dS_{ND}}{dt} = \rho_7 - \rho_6$$

Ammonia is used as the nitrogen source in all growth processes of micro-organisms and the ammonia concentration is affected of all these processes. The fact that ammonia concentration also is decreased by nitrification and increased by ammonification leads to a very complex equation.

$$\frac{dS_{NH}}{dt} = -i_{XB}(\rho_1 + \rho_2) + \rho_6 - \left(i_{XB} + \frac{1}{Y_A}\right)\rho_3$$

The concentration of nitrate is increased by nitrification and decreased by denitrification and this process is described by the ninth equation.

$$\frac{dS_{NO}}{dt} = \frac{1}{Y_A}\rho_3 - \left(\frac{1 - Y_H}{2.86Y_H}\right)\rho_2$$

(The number 2.86 represents the oxygen equivalent for conversion of nitrate nitrogen to nitrogen gas)

The next model equation describes the oxygen concentration in the wastewater. The concentration is decreased by aerobic growth of biomass.

$$\frac{dS_O}{dt} = -\left(\frac{1 - Y_H}{Y_H}\right)\rho_1 - \left(\frac{4.57 - Y_A}{Y_A}\right)\rho_3$$

(The number 4.57 represents the theoretical oxygen consumption for oxidation of ammonium nitrogen to nitrate nitrogen)

The last model equation describes the dynamics of the alkalinity change. The inclusion of alkalinity in the model is to detect problems with changes in pH without including pH in the model (Petersen, 2000).

$$\frac{dS_{ALK}}{dt} = -\frac{i_{XB}}{14}\rho_1 + \left(\frac{1-Y_H}{14.286Y_H} - \frac{i_{XB}}{14}\right)\rho_2 - \left(\frac{2}{14Y_A} + \frac{i_{XB}}{14}\right)\rho_3 + \frac{1}{14}\rho_6$$

In Appendix A the model equations are described in a process matrix.

2.6 Bioreactor Model

The plant consists of 5 microbiological reactors with the first two compartments, k (k=1 and k=2) are the non-aerated compartments with each volume of $V_{as,1}=V_{as,2}=1000m^3$. Compartment 3, 4 and 5 are the aerated compartments (fixed with oxygen transfer coefficient $K_La=10h^{-1}=240d^{-1}$) with each volume is defined as $V_{as,\ 3}=V_{as,4}=V_{as,5}=1333m^3$. At compartment 5, the dissolved oxygen concentration (S_O) is controlled at level of 2 g (-COD).m^3 by manipulating K_La.

2.6.1 Reactor Mass Balance

The general equation for mass balancing:

For k=1 (unit 1)

$$\frac{dZ_1}{dt} = \frac{1}{V_1}(Q_aZ_a + Q_rZ_r + Q_0Z_0 + r_1V_1 - Q_1Z_1)$$

$$Q_1 = Q_a + Q_r + Q_0$$

For k = 2 to 5

$$\frac{dZ_{as,k}}{dt} = \frac{1}{V_{as,k}}\left(Q_{k-1}Z_{as,k} + r_{Z,k}V_{as,k} - Q_kZ_{as,k}\right)$$

$$Q_k = Q_{k-1}$$

For oxygen ($S_{O,as,k}$),

$$\frac{dS_{O,as,k}}{dt} = \frac{1}{V_{as,k}}\left(Q_{k-1}S_{O,as,k-1} + r_{Z,k}V_{as,k} + (K_L a)_k V_{as,k}(S_O^* - S_{O,as,k}) - Q_k S_{O,as,k}\right)$$

Where

S_O^* = 0.8
$r_{Z,k}$ = appropriate conversion rate, depending upon the state variable considered

2.7 Secondary Settler Model

The biomass transformed need to be discharged from the liquid stream before streamed to the receiving waters. This is done in the clarifier where the sedimentation of the biomass particles is achieved by gravity along with the density differences. Partly, the biomass is purged while the rest of it is recycled back to the biological reactor to maintain substrate-to-biomass ratio.

Taking the benchmark as reference (U. Jeppson, 2009) The settler is modelled based on a one dimensional of equally divided 10 layer non-reactive mass balance model taking the 6th layer as the feed input. The settler has an area, A of 1500m^2 and the height of each layer m (z_m) is 0.4m. Therefore the volume is equal to 6000m^3.

Solid flux, $J_S = v_S(X_{SC})$ where X_{SC} is the total sludge concentration. The velocity function is referred from Takacs et al., 1991 is a double exponential:

$$v_{sj} = v_0 e^{-r_h(X_J - X_{min})} - v_0 e^{-r_p(X_J - X_{min})} \text{[m/d]}$$

$$0 \leq v_{sj} \leq v_0^{'}$$

Where

v_0 = maximum theoretical settling velocity in [m/d]
$v_0^{'}$ = maximum partical settling velocity in [m/d]
r_h = the settling parameter characteristic of the hindered settling zone in [m^3/d]
r_p = the settling parameter characteristic of low solids concentration in [m^3/d]
X_{min} = $f_{ns}X_f$, is the minimum attainable suspended solid concentration in [gSS/m^3], with X_{in} is the mixed liquor solids entering the settler and f_{ns} is the non settleable fraction of X_f

Table 2.4: Settling Parameters and its values

Parameter		Value
Maximum settling velocity	v_0'	250.0
Maximum Vesilind settling velocity	v_0	474.0
Hindered zone settling parameter	r_h	0.000576
Flocculant zone settling parameter	r_p	0.00286
Non-settleable fraction	X_f	0.00228

The upward (v_{up}) and downward (v_{dn}) velocities are calculated as:

$$v_{dn} = \frac{Q_u}{A} = \frac{Q_r + Q_w}{A}$$

$$v_{dn} = \frac{Q_e}{A}$$

Therefore the mass balances for the sludge can be written as:

For the feed layer, m=6:

$$\frac{dX_{sc,m}}{dt} = \frac{\frac{Q_f X_f}{A} + J_{sc,m+1} - (v_{up} + v_{dn})X_{sc,m} - \min(J_{s,m}, J_{s,m-1})}{Z_m}$$

For the intermediate layer below the feed layer ($2 \le m \le 5$):

$$\frac{dX_{sc,m}}{dt} = \frac{v_{dn}(X_{sc,m+1} - X_{sc,m}) + \min(J_{s,m}, J_{s,m+1}) - \min(J_{s,m}, J_{s,m-1})}{Z_m}$$

For the bottom layer (m=1):

$$\frac{dX_{sc,1}}{dt} = \frac{v_{dn}(X_{sc,2} - X_{sc,1}) + J_{s,m+1} - J_{s,m}}{Z_1}$$

For the intermediate clarification layers above the feed layer ($7 \le m \le 9$)

$$\frac{dX_{sc,m}}{dt} = \frac{v_{up}(X_{sc,m-1} - X_{sc,m}) + J_{sc,m+1} - J_{sc,m}}{Z_m}$$

$$J_{sc,j} = \begin{cases} \min(v_{s,j}X_{sc,j}, v_{s,j-1}X_{sc,j-1}) \; if \; X_{sc,j-1} > X_t \; or \\ v_{s,j}X_{sc,j} \; if \; X_{sc,j} \le X_t \end{cases}$$

For the top layer (m=10):

$$\frac{dX_{sc,10}}{dt} = \frac{v_{up}\left(X_{sc,9} - X_{sc,10}\right) + J_{sc,10}}{Z_{10}}$$

$$J_{sc,j} = \begin{cases} \min\left(v_{s,10}X_{sc,10}, v_{s,9}X_{sc,9}\right) if\ X_{sc,9} > X_t\ or \\ v_{s,10}X_{sc,10}\ if\ X_{sc,9} \leq X_t \end{cases}$$

The threshold concentration X_t = 3000 g.m^{-3}

For the soluble components which include dissolved oxygen, it is considered that each layer is representing completely mix volume and the concentrations of soluble components are calculated with:

For the feed layer (m=6):

$$\frac{dZ_{sc,m}}{dt} = \frac{\frac{Q_f X_f}{A} - \left(v_{up} + v_{dn}\right)Z_{sc,m}}{Z_m}$$

For the layers below the feed layer (m < 6):

$$\frac{dZ_{sc,m}}{dt} = \frac{v_{dn}\left(Z_{sc,m+1} - Z_{sc,m}\right)}{Z_m}$$

For the layers above the feed layer (m > 6):

$$\frac{dZ_{sc,m}}{dt} = \frac{v_{up}\left(Z_{sc,m-1} - Z_{sc,m}\right)}{Z_m}$$

For recycle and wastage flow, the concentrations are equal to the first layer (bottom layer):

$$z_u = z_{sc,1}$$

For the calculation of the sludge concentration, it is straight forward from the concentrations in compartment 5 of the bioreactor:

$$X_f = \frac{1}{fr_{COD-SS}}(X_{S,as,5} + X_{P,as,5} + X_{I,as,5} + X_{BH,as,5} + X_{BA,as,5})$$

$$X_f = 0.75(X_{S,as,5} + X_{P,as,5} + X_{I,as,5} + X_{BH,as,5} + X_{BA,as,5})$$

Given a COD to SS conversion factor $fr_{\text{COD-SS}}$, is equal to 4/3. The same principle is applied for X_u (in the settler underflow) and X_e (at the plant exit).

To calculate the distribution of particulate concentrations in the recycle and wastage flows, their ratios with respect to the total solid concentration are assumed to remain constant across the secondary clarifier:

$$\frac{X_{S,as,5}}{X_f} = \frac{X_{S,sc,1}}{X_u}$$

It is also applied for $X_{P,sc,1}, X_{I,sc,1}, X_{BH,sc,1}, X_{BA,sc,1}$ and $X_{ND,sc,1}$. It is important to understand that this assumption means that the dynamics of the fractions of particulate concentrations in the inlet of the secondary clarifier will be directly propagated to the secondary clarifier underflow and overflow, without taking into account the normal retention time in the secondary clarifier.

For steady state situation, the sludge age calculation is based on the total amount of biomass present in the plant (the reactor and the settler):

$$SRT = \frac{TX_{as} + TX_{sc}}{\phi_e + \phi_w}$$

Where,

TX_{as} = total amount of biomass present in the reactor (activated sludge)
TX_{sc} = total amount of biomass present in the secondary clarifier
ϕ_e = loss rate of biomass in the secondary clarifier overflow
ϕ_w = loss rate of biomass in the wastage flow

$$TX_{as} = \sum_{j=1}^{k=n} \left(X_{BH,as,k} + X_{BA,as,k}\right).v_{as,k} \; with \; n = 5$$

$$TX_{sc} = \sum_{j=1}^{k=m} \left(X_{BH,sc,j} + X_{BA,sc,j}\right).z_j.A \; with \; m = 10$$

$$\phi_e = \left(X_{BH,sc,m} + X_{BA,sc,m}\right).Q_e$$

$$\phi_w = \left(X_{BH,sc,1} + X_{BA,sc,1}\right).Q_w$$

In the real plant situation, the sludge age is measured on the total amount of solids present in the plant:

$$SRT_{meas} = \frac{TSS_{as} + TSS_{sc}}{\psi_e + \psi_w}$$

Where,

TSS_{as} = total amount of solids present in the reactor (activated sludge)
TSS_{sc} = total amount of solids present in the secondary clarifier
ψ_e = loss rate of solids in the secondary clarifier overflow
ψ_w = loss rate of solids in the wastage flow

$$TSS_{as} = \sum_{j=1}^{k=n} TSS_{as,k} . v_{as,k} \; with \; n = 5$$

Where $TSS_{as,k} = \frac{1}{fr_{COD-SS}} (X_{S,as,i} + X_{P,as,i} + X_{I,as,i} + X_{BH,as,i} + X_{BA,as,i})$

$TSS_{sc} = \sum_{j=1}^{k=m} TSS_{sc,j} . z_j . A \; with \; m = 10$

Where $TSS_{sc,j} = \frac{1}{fr_{COD-SS}} (X_{S,sc,j} + X_{P,sc,j} + X_{I,sc,j} + X_{BH,sc,j} + X_{BA,sc,j})$

$\psi_e = TSS_{sc,m} . Q_e$

Where $TSS_{sc,m} = \frac{1}{fr_{COD-SS}} (X_{S,sc,m} + X_{P,sc,m} + X_{I,sc,m} + X_{BH,sc,m} + X_{BA,sc,m})$

$\psi_w = TSS_{sc,1} . Q_w$

Where $TSS_{sc,1} = \frac{1}{fr_{COD-SS}} (X_{S,sc,1} + X_{P,sc,1} + X_{I,sc,1} + X_{BH,sc,1} + X_{BA,sc,1})$

2.8 Control Strategies

The dissolve oxygen (DO) in the aerobic zone of an activated sludge plays an important role in the whole activated sludge system. The oxygen supply must be sufficient enough for the microorganism to degrade the organic matter and to make sure that ammonium is converted to nitrate. If DO concentration is not sufficient for the growth of microorganism, filamentous organism may predominate and the sedimentation ability and quality of the activated sludge may be poor. On the other hand, excessive supply of oxygen requires a high air flow which would generates large energy consumption and may affect the sludge sedimentation. Besides that,

high DO concentration in the internal recycle flow makes the denitrification process less efficient (Garcia, Sotomayor, & Park, 2002).

The variation of influent loads with the continuous change in respiration rate and oxygen transfer rate causing the DO dynamic to be time-varying. This is quite a challenge to achieve the objective of controlling the DO concentration at a predetermined set-point.

Several control strategies have been suggested in the literature. As a basic strategy, a linear PI controller with feedforward from the respiration rate and the flow rate was presented by Carlsson & Rehnstrom, 2002. Cadet, Beteau, and Carlos Hernandez (2004) have developed a multicriteria control strategy with Takagi–Sugeno fuzzy supervisor system to decrease the total cost although keeping good performances. Besides that, the control of the dissolved oxygen concentration has been approached with good results in the control of a non-linear organic substrate removal process using multimodel techniques (Barbu, Barbu, & Ceanga, 2004).

CHAPTER 3

METHODOLOGY

3.1 General Project Methodology

After the confirmation of the project title, the first thing that is conducted is to understand the wastewater treatment process well. Thorough research has been done to enhance understanding of the project given. This is done through various methods. All resources were made use to gain additional knowledge about the project.

The project is then taken to the next step that is to model and simulate the wastewater treatment plant in which in this case, an activated sludge plant. All the information gathered from the extensive study is applied to model the plant. Matlab/Simulink has been chosen to perform this task and the plant is modelled from the mathematical equation obtained from the literature review. Once completed, plant validation is conducted by comparing the results obtain with similar work conducted by another author using different simulating softwares.

The next objective of this project is to implement control strategies to the plant. First, a basic control strategy, a PI controller as suggested by many in the literature is installed in the plant. Finally, an advance control strategy, model predictive control is to be conducted.

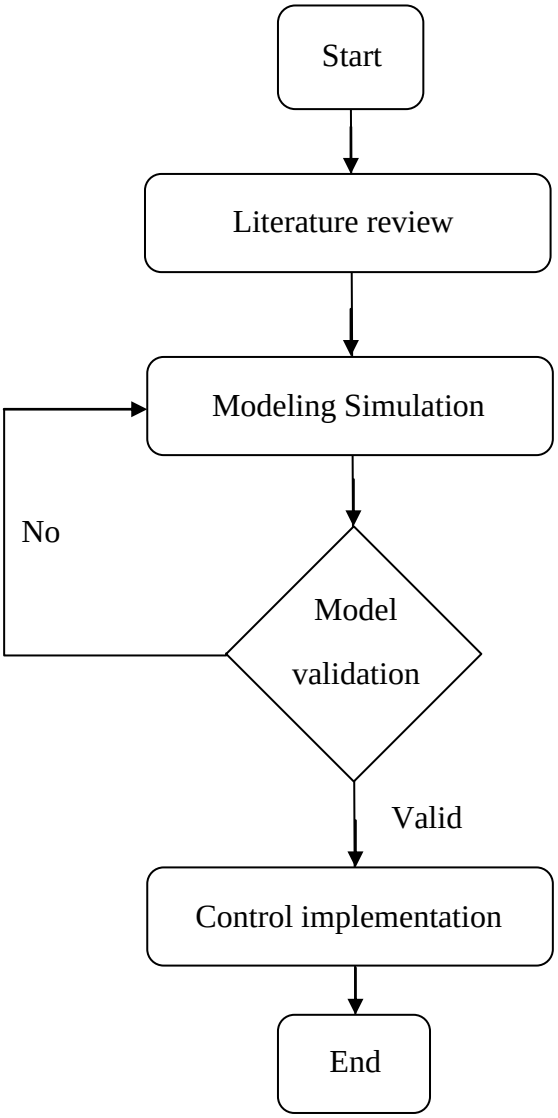

Figure 3.1: Project flow diagram.

3.2 Tool/Software

Simulink is an add-on software product to Matlab for modeling, simulating and analyzing any type of dynamic system. Matlab and Simulink are fully integrated, meaning that all functionalities of the Matlab toolboxes are available in the Simulink environment as well. Simulink provides a graphical user interface for building models as block diagrams and manipulating these blocks dynamically. A large number of predefined building blocks are included and it is easy to extend the functionality by customizing blocks or creating new ones.

The capabilities of Simulink may be further extended by using the S-functions (system functions), which can be written in Matlab language, C++ or Fortran using predefined syntax. Consequently, S-functions can be easily incorporated and a dynamical system can be described as a mathematical set of equations instead of using predefined block diagrams.

For this report, MATLAB/Simulink simulation is selected to conduct the model of activated sludge process

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Open Loop Modeling

An open loop plant is first developed and run for 100 days of dry weather condition with constant values of Q_{int} = 55338 $m^3.d^{-1}$ and $K_La(5)$ = 3.5 h^{-1}. The steady state values are presented in the tables below:

Table 4.1: Biological reactor steady state.

	Influent	k	reac1	reac2	reac3	reac4	reac5	Unit
S_I	30	$S_{I,as,k}$	30	30	30	30	30	$gCOD/m^3$
S_S	69.5	$S_{S\ as,k}$	2.81	1.46	1.15	0.995	0.889	$gCOD/m^3$
X_I	51.2	$X_{I\ as,k}$	1149	1149	1149	1149	1149	$gCOD/m^3$
X_S	202.32	$X_{S\ as,k}$	82.1	76.4	64.9	55.7	49.3	$gCOD/m^3$
X_{BH}	28.17	$X_{BH\ as,k}$	2552	2553	2557	2559	2559	$gCOD/m^3$
X_{BA}	0	$X_{BA\ as,k}$	148.0	148.0	149.0	150.0	150.0	$gCOD/m^3$
X_P	0	$X_{P\ as,k}$	449.0	450.0	450.0	451.0	452.0	$gCOD/m^3$
S_O	0	$S_{O\ as,k}$	0.00430	0.000631	1.72	2.43	0.491	gO_2/m^3
S_{NO}	0	$S_{NO\ as,k}$	5.37	3.66	6.54	9.30	10.4	gN/m^3
S_{NH}	31.56	$S_{NH\ as,k}$	7.92	8.34	5.55	2.97	1.73	gN/m^3
S_{ND}	6.95	$S_{ND\ as,k}$	1.22	0.882	0.829	0.767	0.688	gN/m^3
X_{ND}	10.59	$X_{ND\ as,k}$	5.28	5.03	4.39	3.88	3.53	gN/m^3
S_{ALK}	7	$S_{AL\ as,k\ K}$	4.93	5.08	4.67	4.29	4.13	$mole.m^{-3}$
TSS		$TS_{as,k}$ S	3285	3282	3278	3274	3270	G $SS.m^{-3}$
Q_i	18446	Q_k	92230	92230	92230	92230	92230	$m^3.d^{-1}$

Table 4.2: Concentration of solids and soluble components in settler (steady state).

	$TSS_{sc,k}$	$SI_{sc,k}$	$SS_{sc,k}$	$SO_{sc,k}$	$SNO_{sc,k}$	$S_{NHsc,k}$	$S_{NDsc,k}$	$S_{ALKsc,k}$
10	12.5	30	0.889	0.491	10.4	1.73	0.688	4.13
9	18.1	30	0.889	0.491	10.4	1.73	0.688	4.13
8	29.5	30	0.889	0.491	10.4	1.73	0.688	4.13
7	69.0	30	0.889	0.491	10.4	1.73	0.688	4.13
6	356.0	30	0.889	0.491	10.4	1.73	0.688	4.13
5	356.0	30	0.889	0.491	10.4	1.73	0.688	4.13
4	356.0	30	0.889	0.491	10.4	1.73	0.688	4.13
3	356.0	30	0.889	0.491	10.4	1.73	0.688	4.13
2	356.0	30	0.889	0.491	10.4	1.73	0.688	4.13
1	6394.0	30	0.889	0.491	10.4	1.73	0.688	4.13

With this steady state values as initial parameters to the plant, the plant is simulated for 14 days. The integral calculations are based on the last 7 days. Therefore, the first data should be the one at time 7.00 day and the last data should be the one at time 13.99 day. The data at time 14 day should not be included in the integral since forward Euler approximation is used.

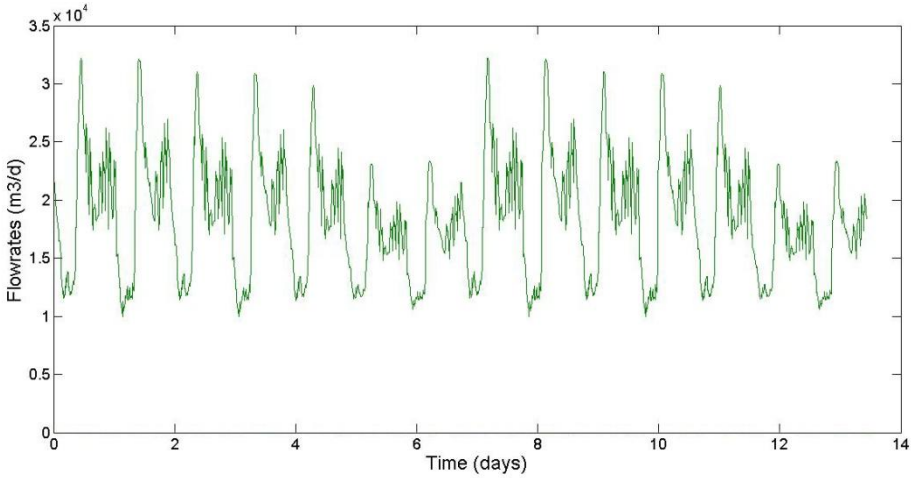

Figure 4.1: Influent load.

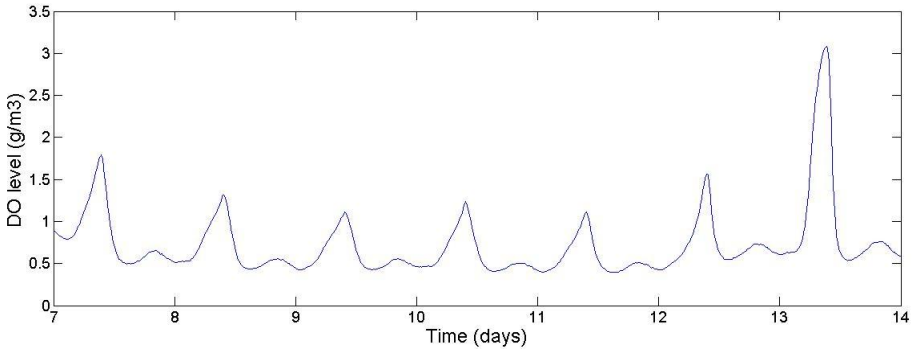

Figure 4.2: Dissolve oxygen concentration after simulation of steady state for 14 days.

Table 4.3: Model comparison with similar work using GPS-X and SIMBA software. (Copp, 2002)

Effluent average concentrations based on load	Unit	Results generated	GPS-X	SIMBA
S_I	$gCOD/m^3$	29.9789	30.0000	30.0000
S_S	$gCOD/m^3$	0.9470	0.9694	0.9735
X_I	$gCOD/m^3$	4.5779	4.5878	4.5795
X_S	$gCOD/m^3$	0.3190	0.2250	0.2229
X_{BH}	$gCOD/m^3$	10.0892	10.2219	10.2209
X_{BA}	$gCOD/m^3$	0.5458	0.5412	0.5422
X_P	$gCOD/m^3$	1.7560	1.7580	1.7572
S_O	gO_2/m^3	0.7308	0.7978	0.7463
S_{NO}	gN/m^3	8.8131	8.8464	8.8237
S_{NH}	gN/m^3	4.7632	4.8571	4.7590
S_{ND}	gN/m^3	0.7591	0.7260	0.7290
X_{ND}	gN/m^3	0.0122	0.0158	0.0157
S_{ALK}	Molar units	5.3183	n/a	4.4562
TSS	gSS/m^3	11.5695	13.0004	12.9919

From Table 4.3, the results generated did not vary much with literature. It is safe to say that the plant model is reliable enough to proceed with control implementation.

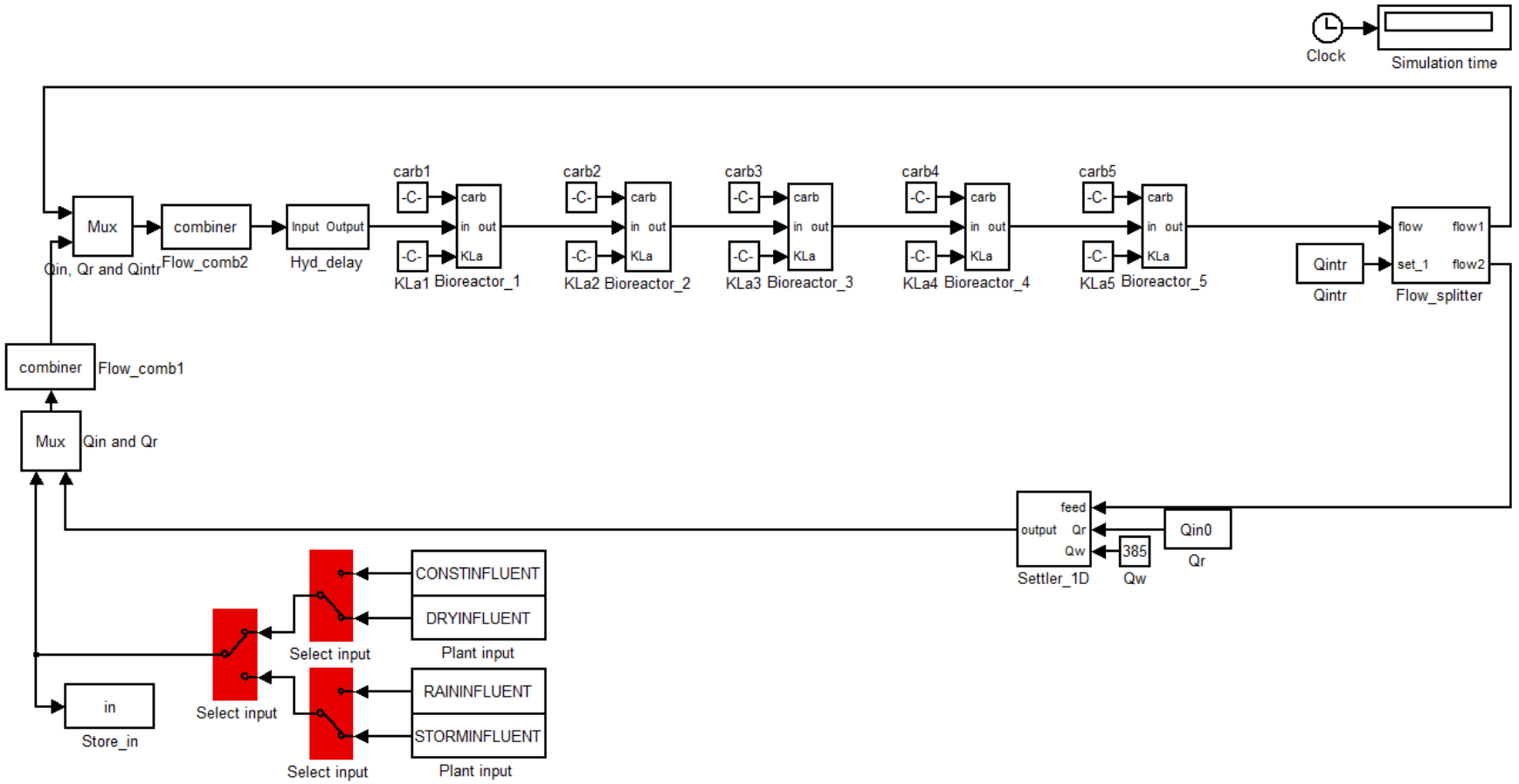

Figure 4.3: Open loop model. (Jeppson, 1996)

4.2 Close Loop Modeling

Once the steady state model is set up, basic control strategy can be implemented to the plant. In this research, the control of dissolve oxygen is of interest for its crucial impact on the plant performance as well as to the financial effect of the plant as discussed in the literature.

Thus, the aim of this controller is to control dissolved oxygen concentration in the fifth compartment at a predetermined set point value (2 g(-COD).m^{-3}). The manipulated variable is the oxygen transfer coefficient, K_La. There are also constraints on oxygen transfer which is 0 to 10 h^{-1}. At influent, the value of dissolve oxygen (SO) is set at 0 g.COD/m^3.

The performance of the controller is assessed by

- Integral of Absolute Error (IAE)

$$IAE = \int_{t_o}^{t_f} |e_i| . dt$$

- Integral of Squared Error (ISE)

$$ISE = \int_{t_o}^{t_f} e_i^2 . dt$$

- maximal deviation from set point

$$Dev_i^{max} = max\{e_i\}$$

- Variance of manipulated variable (u_i) variations.

$$Var(\Delta u_i) = \overline{\Delta u_i^2} - (\overline{\Delta u_i})^2$$

The evaluation period for each controller spreads over the last seven days of the simulation. Tuning has been performed for minimizing the ISE criterion and was achieved by repeated simulations with gain, K value of integral time constant of 0.0023 and as suggested by Copp, 2002. (See Appendix D for simulink model of the controller)

Table 4.4: Evaluation performance.

Controller type	PI
Set point	2.000
Integral of absolute error (IAE)	0.0177
Integral of square error (ISE)	0.03
Max abs deviation from setpoint (max e)	0.2251
Variance of error (var e)	0.0629

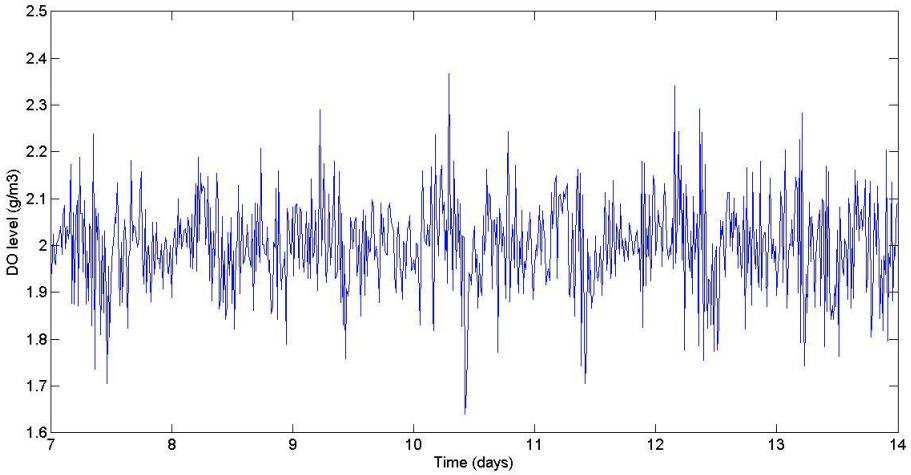

Figure 4.4: Dissolve oxygen level with PI controller (setpoint = 2gCOD/m^3).

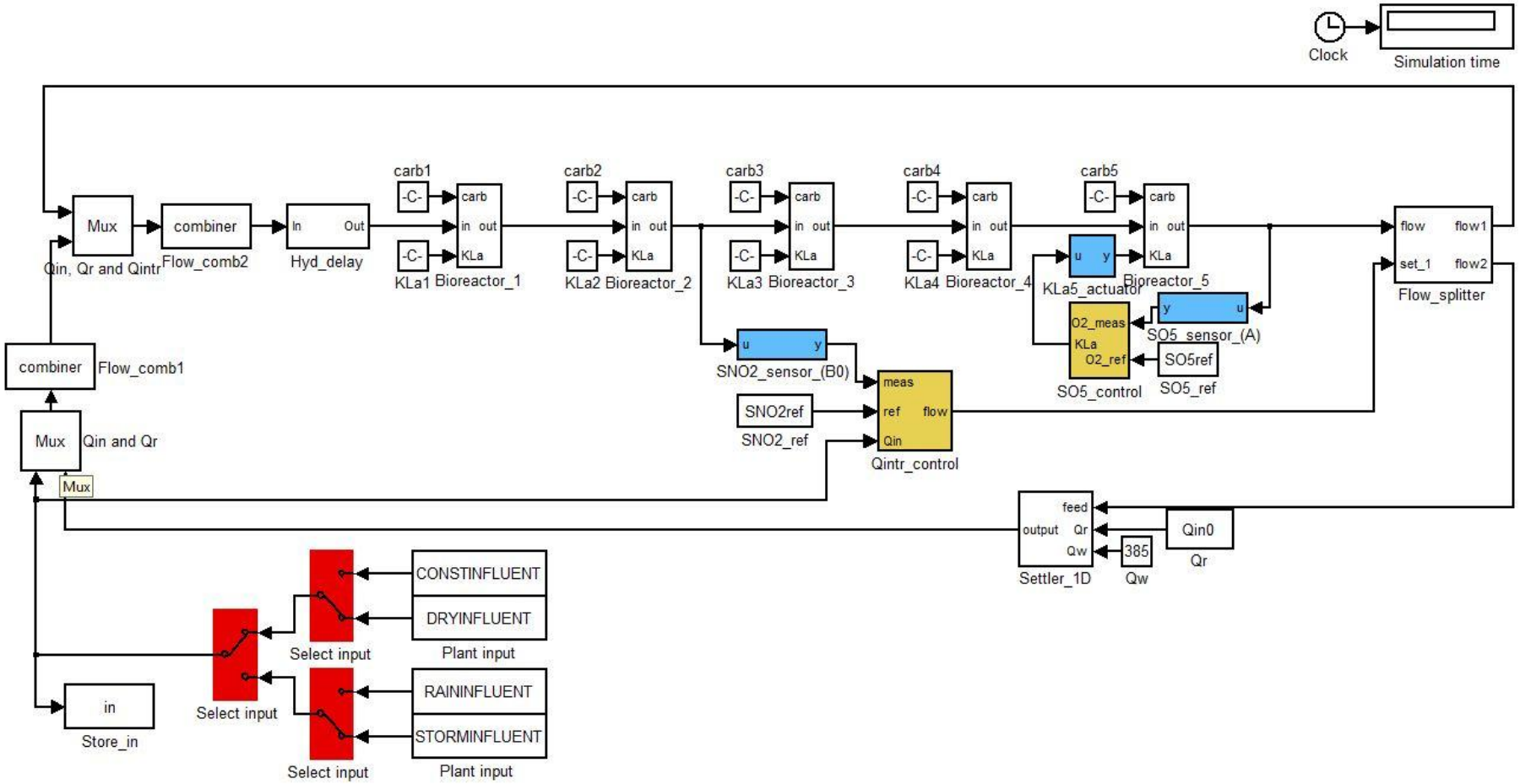

Figure 4.5: Close loop model.

4.3 Model Predictive Control (MPC)

Model predictive control (MPC) is an advance control algorithm to predict future plant behaviour by computing a series of manipulated variables adjustments. The first input in the optimal sequence is then sent into the plant, and the entire calculation is repeated at subsequent control intervals. (Figure 4.8)

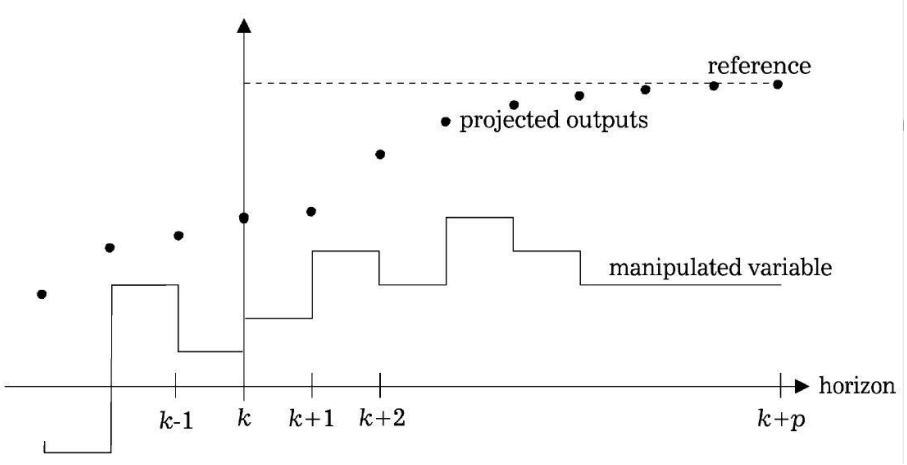

Figure 4.6: Model predictive control.

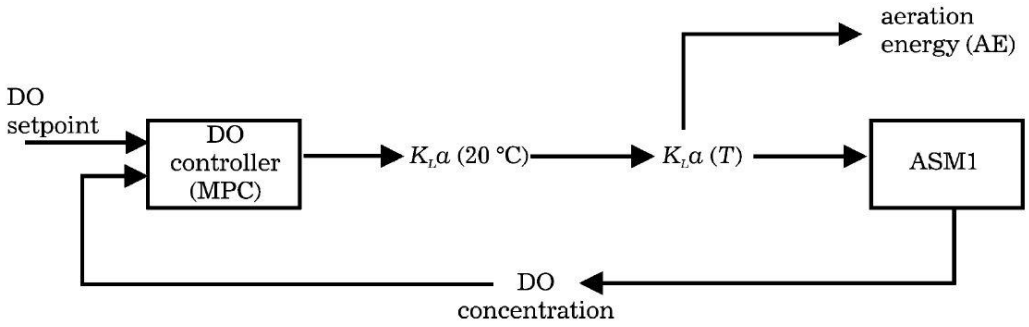

Figure 4.7: Schematic view of the dissolve oxygen control process.

4.3.1 System Identification

The first step of MPC is to obtain the process model. This can be achieved through conducting system identification which is a technique to obtain process model through testing input-output data. In this process, the concentrations of the DO at reactor 5 (y_1) and reactor 2 (y_2) are considered as the plant output and the manipulated variable is the oxygen mass transfer coefficient (u_1) and internal recycle

flowrate (u_2). Internal recycle flow rate influences the supply of nitrate for the denitrification process but also the DO concentration in the anoxic reactors, since DO may be also transported from the aerated reactors. First, system identification is conducted using MATLAB toolbox to obtain the process model.

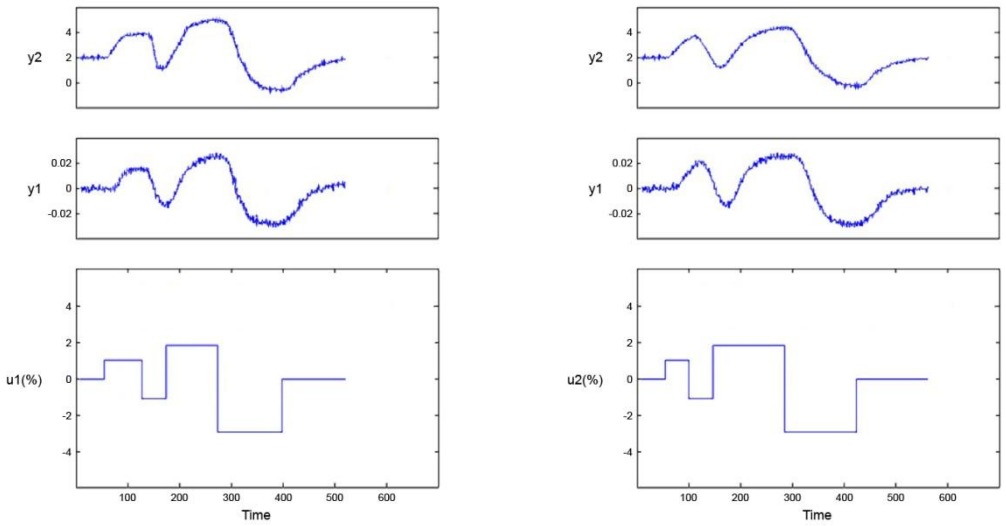

Figure 4.8: Step testing of input-output of y_1 and y_2 corresponding to u_1 and u_2.

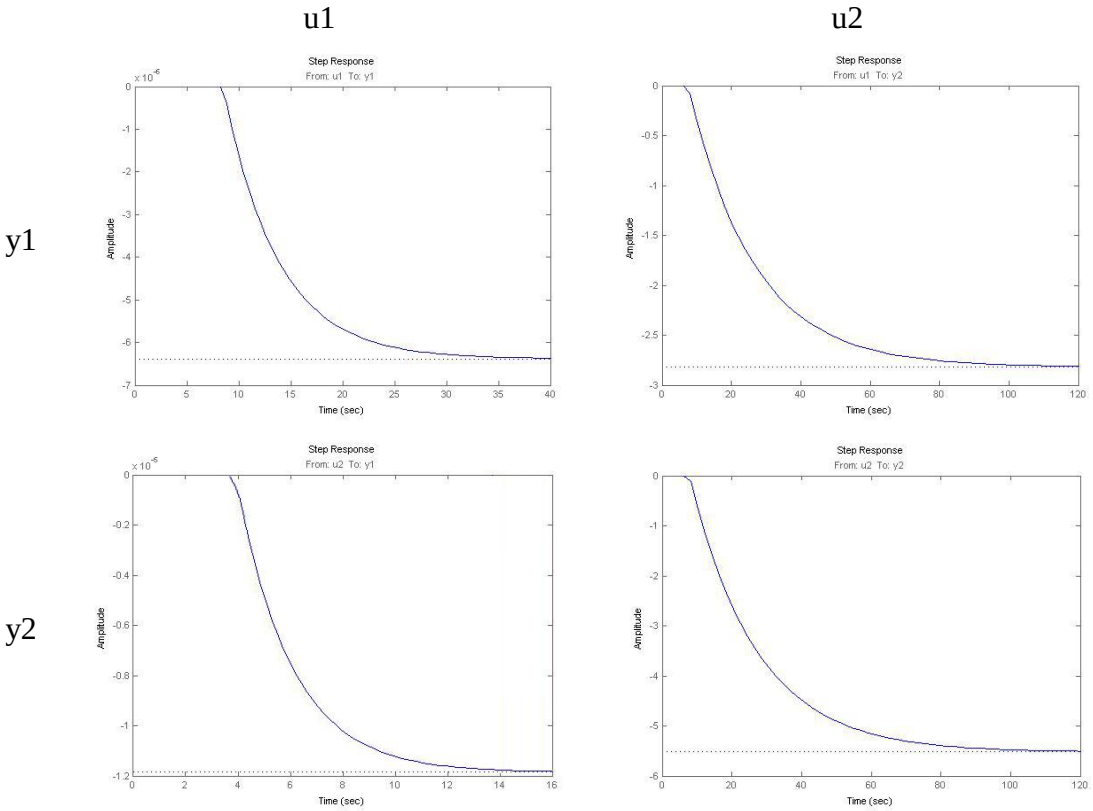

Figure 4.9: Step response of process model.

Table 4.5: Process model values.

Transfer function	K (h^{-1})	Tp (sec)
g11	-6.3812e-6	5.202
g12	-2.8177	18.898
g21	-1.1849e-005	3.267
g22	-5.5206	19.212

From the result obtain, the gain values are negative which describe the behaviour as an increase in the controlled variable (output) y requires an increase in the manipulative variables.

4.3.2 Setpoint Tracking

The smoothness of the process from its nominal values to its output can be measured by setpoint tracking. One case is studied whereby output y_1 is stepped up with the nominal values are as shown in Table 4.6. The response of the control is presented in Figure 4.10.

Table 4.6: Nominal input and output values.

Index	Input (%)	Output (g/m^3)
1	43	1.4
2	61	0.5

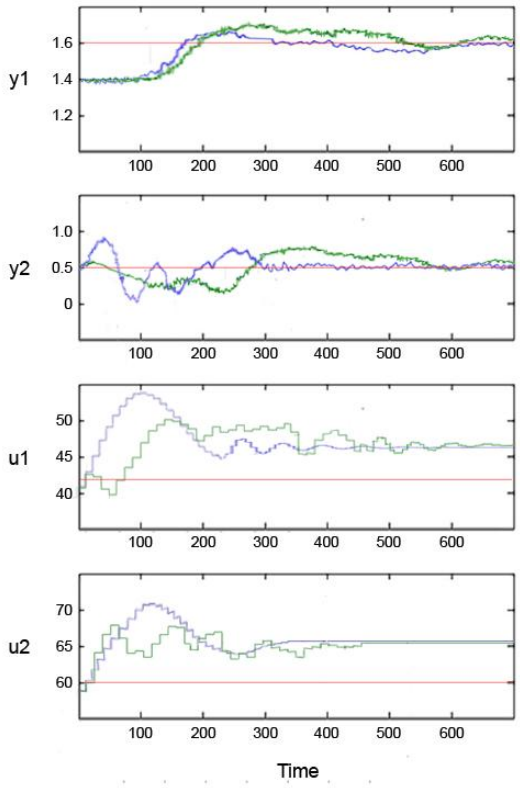

Figure 4.10: Setpoint tracking in increment of output y_1 with blue line represent the PI controller response, green line is the MPC and red line is the setpoint.

Table 4.7: ISE values of different set point.

Output	ISE	
	MPC	PI
y_1	15.43	15.76
y_2	7.02	7.49

CHAPTER 5

CONCLUSION AND RECOMMENDATION

At the end of this work, the objectives are achieved. An activated sludge plant is modelled and the open loop datum validated to be in close accuracy against literature. The activated sludge plant had also been implemented with PI control and MPC. It is shown in general that MPC is possible to improve PI control with more precise tuning to increase the efficiency of the MPC control.

Certain complication of the system has been encountered which requires further analysis to the system. The large complicated model makes the analysis become harder. For future work, it is recommended to conduct a model reduction process to reduce its complexity (Ilse, Jeroen, Ronald, & Jan, 2003). The model can also be reduced using a singular perturbation method for easier plant analysis (David, Wouwer, Queinnec, & Vasel, 2009).

References

1. Abd. Wahab, N., Katebi, R., & Balderud, J. (2009). Multivariable PID control design for activated sludge process with nitrification and denitrification. Biochemical Engineering Journal 45 , 239-248.

2. Alex J., Benedetti L., Copp J., Gernaey K.V., Jeppsson U., Nopens I., Pons M.N., Rieger L., Rosen C., Steyer J.P., Vanrolleghem P., Winkler S., 2008. Benchmark Simulation Model No. 1 (BSM1). Dept. of Industrial Electrical Engineering and Automation, Lund University

3. Alsina, X. F. (2008). Conceptual Design of wastewater treatment plant using multiple objectives.

4. Araujo, A., Gallani, S., Mulas, M., & Olsson, G. (2011). A systematic approach to the design of operation and control policies in activated sludge systems. Published by American Chemical Society.

5. Balazs Holenda, Doctorate School of Chemical Engineering, University of Pannonia, 2007, Development of Modeling, Control and Optimization Tools for the Activated Sludge process. (1-142).

6. Barbu, M., & Caraman, S. (2007). QFT Multivariabil Control of a Biotechnological. Proceedings of 10th IFAC Symposium on Computer Applications in Biotechnology. Cancun.

7. Barbu, M., Barbu, G., & Ceanga, E. (2004). The Multi-model Control of the Wastewater Treatment Process with Activated Sludge. Proceedings of 12th Mediterranean Conference on Control and Automation - MED'04. Kusadasi, Turkey.

8. Cadet, C., Beteau, J., & Carlos Hernandez, S. (2004). Multicriteria control strategy for cost/quality compromise in Wastewater Treatment Plants. Control Engineering Practice, 12 , 335-347.

9. Cajsa Hellstedt, Department of Information Technology, Uppsala University, (2005), Calibration of a dynamic model for the activeted sludge process at Henriksdal wastewater treatment plant.

10. Carlsson, B., & Rehnstorm, A. (2002). Control of an activated sludge process with nitrogen removal-a benchmark study. Water Science and Technology, 45(4/5) , 135-142.

11. Chachuat, B., Roche, N., & Latifi, M. (2005). Optimal aeration control of industrial alternating activated sludge plants. Biochemical Engineering Journal 23 , 277-289.

12. Chai, Q. (2008). Modeling, estimation, and control of biological wastewater treatment plants. Norway.

13. Chai, Q., & Lie, B. (June 11-13, 2008). Predictive Control of an Intermittently Aerated Activated Sludge Process. American Control Conference, (pp. 2209-2214). Washington, USA.

14. Chen, J., Gao, Y., Zhang, Z., Shi, D., & Xi, D. (7-11 August,2006). Integrated modeling and simulation of activated sludge process. Proceedings of the 25th Chinese Control Conference, (pp. 1382-1386). Harbin, Heilongjiang.

15. Copp, J. (1998). The COST Simulation Benchmark: Description and Simulator Manual.

16. Copp, J. (2002). The COST simulation benchmark: Description and simulator manual.

17. David, R., Vasel, J.-L., & Wouwer, A. V. (2006). Settler dynamic modeling and MATLAB simulation of the activated sludge process. Chemical Engineering Journal 146 , 174-183.

18. David, R., Wouwer, A. V., Queinnec, I., & Vasel, J.-L. (2009). Robust Control of the Activated Sludge Process. American Institute of Chemical Engineers Biotechnol. Prog., 25 , 701-708.

19. Ekama G.A., Barnard J.L., Gunthert F.W., Krebs P., McConquodale J.A., Parker D.S., Wahalberg E.J. 1997. *Secondary settling tanks: theory, modeling, design and operation*. Technical Report 6, IWA Publishing.

20. Ekman, M., Bjorlenius, B., & Andersson, M. (2006). Control of the aeration volume in an activated sludge process using supervisory control strategies. WATE R RESEARCH 40 , 1668-1676.

21. Environmental Quality Act 1974, Environmental Quality (Sewage, and Industrial Effluents Regulation 1978), (pp. 60), KL, DOE.

22. Garcia, C., Sotomayor, O., & Park, S. (2002). A referencemodel for evaluating control strategies in activated sludge wastewater treatment plants. Control Stategies in Wastewater Treatment , 543-556.

23. Grau P., Copp J., Vanrolleghem P.A., Takács I., Ayesa E. 2009. A comparative analysis of different approaches for integrated WWTP modeling. *Water Science and Technology*, 59(1),141-147.

24. Halvarsson, B. (2010). Interaction Analysis in Multivariable Control Systems. Applications to Bioreactors for Nitrogen Removal. Acta

Universitatis Upsaliensis. Uppsala Dissertations from the Faculty of Science and Technology. Uppsala.

25. Henri Haimi, Michela Mulas, Kristian Sahlstedt and Riku Vahala, 2009, Helsinki University of Technology Water and Wastewater Engineering, Advanced Operation and Control Methods of Municipal Wastewater Treatment Processes in Finland, 1-83.

26. Henze, M., Grady, C., Jr., P., Gujer, W., Marais, G., & Matsuo, T. (1987). Activated sludge model no.1, IAWPRC Scientific and Technical Report No. 1,. London.

27. Henze M., Gujer W., Mino T., van Loosedrecht M. 2002. *Activated Sludge Models ASM1, ASM2, ASM2d and ASM3*. Scientific and Technical Reports, No. 9, IWA Publishing.

28. Henze M., van Loosdrecht M.C.C, Brdjanovic D., Ekama G.A. 2008. *Biological Wastewater Treatment: Principles, Modeling and Design*. IWA Publishing.

29. I. Diaz, A,C, Lopes, S.i. Perez, M. Fds.-Polanco, (2010), Performance Evaluation of Oxygen, air and nitrate for the microaerobic removal of hydrogen sulphide in biogas from sludge digestion, Bioresource Technology 101.

30. Ilse, Y. S., Jeroen, V. H., Ronald, C., & Jan, F. V. (2003). Linearization of the activated sludge model ASM1 for fast and reliable predictions. Water Research 37 , 1831-1851.

31. Ishak, A. A., Hussain, M. A., & Elkanzi, E. (2001). MODELING AND CONTROL STUDIES OF WASTEWATER NEUTRALIZATION PROCESS. Proceeding to Brunei International Conference on Engineering & Technology, (pp. 221-231). Bandar Seri Begawan.

32. Jeppson U., (1996), *Modeling aspects of wastewater treatment processes*, Department of industrial electrical engineering and automation, Lund institute of technology.

33. Jeppsson U. Alex J., Pons M.N., Spanjers H. and Vanrolleghem P.A., 2002. Status and future trends of ICA in wastewater treatment – a European perspective. *Water Science and Technology*, 45(45), 485-494.

34. Jeppsson, U. and Pons, M.N., 2004. Editorial: The COST benchmark simulation model – current state and future perspective. *Control Engineering Practice*, **12**(3), 299–304.

35. JP Maree, HA Greben and M de Beer (2004), Treatment of acid and sulfide rich effluent in an integrated biological/chemical process, Division of Water, Environment and Forestry Technology, 30(2), pg 183-190.

36. Katebi, M., & Sanchez, A. PREDICTIVE CONTROL OF DISSOLVED OXYGEN IN AN ACTIVATED SLUDGEWASTEWATER TREATMENT PLANT. Scotland.

37. Mario, F., Hernan, A., Pastora, V., & Revollar, S. (2010). Stable Model Predictive Control tuning considering asymmetric bounded signals. Elsevier B.V.

38. Metcalf and Eddy (2004), Wastewater Engineering Treatment and Reuse, (pp. 1-139, 547-565), CA, Mc Graw Hill.

39. Mulas, M. (2006). Modeling and Control of Activated Sludge Process. Cagliari.

40. Petersen, B. (2000). Calibration, identifiability and optimal experimental design of activated sludge models.

41. Queinnec I., Docchain D. 2001, Modeling and simulation of the steady-state of secondary settlers in wastewater treatment plants, Water Science and technology, 43(7), 39-46.

42. Rocca, J., & Yang, S. X. (August 2009). Fuzzy Control of Dissolved Oxygen Concentration in a Bioreactor for Wastewater Applications. Proceedings of the IEEE International Conference on Automation and Logistics, (pp. 108-113). Shenyang, China.

43. V. Erbe, L.P. Risholt, W. Schilling, J. Londong, Integrated modeling for analysis and optimisation of wastewater systems – the Odenthal case, Urban Water 4 (2002) 63.

44. Yong, M., Yongzhen, P., & Jeppsson, U. (2006). Dynamic evaluation of integrated control strategies for enhanced nitrogen removal in activated sludge processes. Control Engineering Practice 14 , 1269-1278.

Appendix A - ASM1 process matrix (Henze et. al., 2002)

Component (i) → ↓ Process (j)	1 S_I	2 S_S	3 X_I	4 X_S	5 X_{BH}	6 X_{BA}	7 X_P	8 S_O	9 S_{NO}	10 S_{NH}	11 S_{ND}	12 X_{ND}	13 S_{ALK}	Process rate (ρ_j)
1 Aerobic growth of heterotrophic biomass		$-\frac{1}{Y_H}$			1			$-\frac{1-Y_H}{Y_H}$		$-i_{XB}$			$\frac{i_{XB}}{14}$	$\mu_{maxH}\frac{S_S}{K_S+S_S}\frac{S_O}{K_{OH}+S_O}X_{BH}$
2 Anoxic growth of heterotrophic biomass		$-\frac{1}{Y_H}$			1					$-i_{XB}$			$\frac{1-Y_H}{14\cdot 2.86Y_H}$ $-\frac{i_{XB}}{14}$	$\eta_g\mu_{maxH}\frac{S_S}{K_S+S_S}\frac{K_{OH}}{K_{OH}+S_O}\frac{S_{NO}}{K_{NO}+S_{NO}}X_{BH}$
3 Aerobic growth of autotrophic biomass						1		$-\frac{4.57-Y_A}{Y_A}$	$\frac{1}{Y_A}$	$-i_{XB}-\frac{1}{Y_A}$			$\frac{2}{14Y_A}-\frac{i_{XB}}{14}$	$\mu_{maxA}\frac{S_{NH}}{K_{NH}+S_{NH}}\frac{SO}{K_{OA}+SO}X_{BA}$
4 Decay of heterotrophic biomass				$1-f_P$	-1		f_P					$i_{XB}-f_P i_{XB}$		$b_H X_{BH}$
5 Decay of autotrophic biomass				$1-f_P$		-1	f_P					$i_{XB}-f_P i_{XB}$		$b_A X_{BA}$
6 Ammonification of soluble organic nitrogen										1	-1		$\frac{1}{14}$	$k_a S_{ND} X_{BH}$
7 Hydrolysis of slowly biodegradable substrate		1		-1										$k_h\frac{X_S/X_{BH}}{K_X+X_S/X_{BH}}\frac{S_O}{K_{OH}+S_O}+\eta_h\frac{K_{OH}}{K_{OH}+S_O}\frac{S_{NO}}{K_{NO}+S_{NO}}X_{BH}$
8 Hydrolysis of organic nitrogen											1	-1		$\rho_7(X_{ND}/X_S)$

Appendix B – C file S function ASM1

```
/*
 * ASM1 is a C-file S-function for AS Model No 1.
 *
 */

#define S_FUNCTION_NAME asm1

#include "simstruc.h"
#include <math.h>

#define XINIT   ssGetArg(S,0)
#define PAR ssGetArg(S,1)
#define V   ssGetArg(S,2)
#define SOSAT   ssGetArg(S,3)

/*
 * mdlInitializeSizes - initialize the sizes array
 */
static void mdlInitializeSizes(SimStruct *S)
{
    ssSetNumContStates(    S, 13); /*13 number of continuous states
*/
    ssSetNumDiscStates(    S, 0);   /* number of discrete states
*/
    ssSetNumInputs(        S, 16);   /* number of inputs
*/
    ssSetNumOutputs(       S, 15);   /* number of outputs
*/
    ssSetDirectFeedThrough(S, 1);   /* direct feedthrough flag
*/
    ssSetNumSampleTimes(   S, 1);   /* number of sample times
*/
    ssSetNumSFcnParams(    S, 4);   /* number of input arguments
*/
    ssSetNumRWork(         S, 0);   /* number of real work vector
elements   */
    ssSetNumIWork(         S, 0);   /* number of integer work vector
elements*/
    ssSetNumPWork(         S, 0);   /* number of pointer work vector
elements*/
}

/*
 * mdlInitializeSampleTimes - initialize the sample times array
 */
static void mdlInitializeSampleTimes(SimStruct *S)
{
    ssSetSampleTime(S, 0, CONTINUOUS_SAMPLE_TIME);
    ssSetOffsetTime(S, 0, 0.0);
}



/*
 * mdlInitializeConditions - initialize the states
 */
static void mdlInitializeConditions(double *x0, SimStruct *S)
{
```

```
int i;

for (i = 0; i < 13; i++) {
   x0[i] = mxGetPr(XINIT)[i];
}
}

/*
 * mdlOutputs - compute the outputs
 */

static void mdlOutputs(double *y, double *x, double *u, SimStruct
*S, int tid)
{
  double X_I2TSS, X_S2TSS, X_BH2TSS, X_BA2TSS, X_P2TSS;
  int i;

  X_I2TSS = mxGetPr(PAR)[19];
  X_S2TSS = mxGetPr(PAR)[20];
  X_BH2TSS = mxGetPr(PAR)[21];
  X_BA2TSS = mxGetPr(PAR)[22];
  X_P2TSS = mxGetPr(PAR)[23];

  for (i = 0; i < 13; i++) {
      y[i] = x[i];
  }


y[13]=X_I2TSS*x[2]+X_S2TSS*x[3]+X_BH2TSS*x[4]+X_BA2TSS*x[5]+X_P2TSS*
x[6];
  y[14]=u[14];  /* x[13] */

}

/*
 * mdlUpdate - perform action at major integration time step
 */

static void mdlUpdate(double *x, double *u, SimStruct *S, int tid)
{
}

/*
 * mdlDerivatives - compute the derivatives
 */
static void mdlDerivatives(double *dx, double *x, double *u,
SimStruct *S, int tid)
{

double mu_H, K_S, K_OH, K_NO, b_H, mu_A, K_NH, K_OA, b_A, ny_g, k_a,
k_h, K_X, ny_h;
double Y_H, Y_A, f_P, i_XB, i_XP;
double proc1, proc2, proc3, proc4, proc5, proc6, proc7, proc8,
proc3x;
double reac1, reac2, reac3, reac4, reac5, reac6, reac7, reac8,
reac9, reac10, reac11, reac12, reac13;
double vol, SO_sat, T;
double xtemp[13];
int i;
```

```
mu_H = mxGetPr(PAR)[0];
K_S = mxGetPr(PAR)[1];
K_OH = mxGetPr(PAR)[2];
K_NO = mxGetPr(PAR)[3];
b_H = mxGetPr(PAR)[4];
mu_A = mxGetPr(PAR)[5];
K_NH = mxGetPr(PAR)[6];
K_OA = mxGetPr(PAR)[7];
b_A = mxGetPr(PAR)[8];
ny_g = mxGetPr(PAR)[9];
k_a = mxGetPr(PAR)[10];
k_h = mxGetPr(PAR)[11];
K_X = mxGetPr(PAR)[12];
ny_h = mxGetPr(PAR)[13];
Y_H = mxGetPr(PAR)[14];
Y_A = mxGetPr(PAR)[15];
f_P = mxGetPr(PAR)[16];
i_XB = mxGetPr(PAR)[17];
i_XP = mxGetPr(PAR)[18];
vol = mxGetPr(V)[0];
SO_sat = mxGetPr(SOSAT)[0];

for (i = 0; i < 13; i++) {
   if (x[i] < 0.0)
     xtemp[i] = 0.0;
   else
     xtemp[i] = x[i];
}

if (u[15] < 0.0)
      x[7] = fabs(u[15]);

proc1 =
mu_H*(xtemp[1]/(K_S+xtemp[1]))*(xtemp[7]/(K_OH+xtemp[7]))*xtemp[4];
proc2 =
mu_H*(xtemp[1]/(K_S+xtemp[1]))*(K_OH/(K_OH+xtemp[7]))*(xtemp[8]/(K_N
O+xtemp[8]))*ny_g*xtemp[4];
proc3 =
mu_A*(xtemp[9]/(K_NH+xtemp[9]))*(xtemp[7]/(K_OA+xtemp[7]))*xtemp[5];
/* in GPS-X they use proc3x instead of proc3 in the oxygen equation
*/
/* proc3x =
mu_A*(xtemp[9]/(K_NH+xtemp[9]))*(xtemp[7]/(K_OH+xtemp[7]))*xtemp[5];
*/
proc4 = b_H*xtemp[4];
proc5 = b_A*xtemp[5];
proc6 = k_a*xtemp[10]*xtemp[4];
proc7 =
k_h*((xtemp[3]/xtemp[4])/(K_X+(xtemp[3]/xtemp[4])))*((xtemp[7]/(K_OH
+xtemp[7]))+ny_h*(K_OH/(K_OH+xtemp[7]))*(xtemp[8]/(K_NO+xtemp[8])))*
xtemp[4];
proc8 = proc7*xtemp[11]/xtemp[3];

reac1 = 0.0;
reac2 = (-proc1-proc2)/Y_H+proc7;
reac3 = 0.0;
reac4 = (1.0-f_P)*(proc4+proc5)-proc7;
reac5 = proc1+proc2-proc4;
reac6 = proc3-proc5;
reac7 = f_P*(proc4+proc5);
```

```
reac8 = -((1.0-Y_H)/Y_H)*proc1-((4.57-Y_A)/Y_A)*proc3;
reac9 = -((1.0-Y_H)/(2.86*Y_H))*proc2+proc3/Y_A;
reac10 = -i_XB*(proc1+proc2)-(i_XB+(1.0/Y_A))*proc3+proc6;
reac11 = -proc6+proc8;
reac12 = (i_XB-f_P*i_XP)*(proc4+proc5)-proc8;
reac13 = -i_XB/14*proc1+((1.0-Y_H)/(14.0*2.86*Y_H)-
(i_XB/14.0))*proc2-((i_XB/14.0)+1.0/(7.0*Y_A))*proc3+proc6/14;

dx[0] = 1.0/vol*(u[14]*(u[0]-x[0]))+reac1;
dx[1] = 1.0/vol*(u[14]*(u[1]-x[1]))+reac2;
dx[2] = 1.0/vol*(u[14]*(u[2]-x[2]))+reac3;
dx[3] = 1.0/vol*(u[14]*(u[3]-x[3]))+reac4;
dx[4] = 1.0/vol*(u[14]*(u[4]-x[4]))+reac5;
dx[5] = 1.0/vol*(u[14]*(u[5]-x[5]))+reac6;
dx[6] = 1.0/vol*(u[14]*(u[6]-x[6]))+reac7;
if (u[15] < 0.0)
      dx[7] = 0.0;
   else
      dx[7] = 1.0/vol*(u[14]*(u[7]-x[7]))+reac8+u[15]*(SO_sat-x[7]);
dx[8] = 1.0/vol*(u[14]*(u[8]-x[8]))+reac9;
dx[9] = 1.0/vol*(u[14]*(u[9]-x[9]))+reac10;
dx[10] = 1.0/vol*(u[14]*(u[10]-x[10]))+reac11;
dx[11] = 1.0/vol*(u[14]*(u[11]-x[11]))+reac12;
dx[12] = 1.0/vol*(u[14]*(u[12]-x[12]))+reac13;
/*dx[13] = (u[14]-x[13])/T;   low pass filter for flow, avoid
algebraic loops */

}


/*
 * mdlTerminate - called when the simulation is terminated.
 */
static void mdlTerminate(SimStruct *S)
{
}

#ifdef  MATLAB_MEX_FILE    /* Is this file being compiled as a MEX-
file? */
#include "simulink.c"      /* MEX-file interface mechanism */
#else
#include "cg_sfun.h"       /* Code generation registration function
*/
#endif
```

Appendix C – Settler model C file S function

```
/*
 * SETTLER1D is a C-file S-function for defining a 10 layer settler
model.
 * can simulate 0, 1 or 10 layers for the solubles by using
MODELTYPE
 *
 */

#define S_FUNCTION_NAME settler1dv4

#include "simstruc.h"
#include <math.h>

#define XINIT   ssGetArg(S,0)
#define PAR ssGetArg(S,1)
#define DIM ssGetArg(S,2)
#define LAYER   ssGetArg(S,3)
#define MODELTYPE   ssGetArg(S,4)

/*
 * mdlInitializeSizes - initialize the sizes array
 */
static void mdlInitializeSizes(SimStruct *S)
{
    ssSetNumContStates(    S, 80);   /* number of continuous states
*/
    ssSetNumDiscStates(    S, 0);   /* number of discrete states
*/
    ssSetNumInputs(        S, 17);   /* number of inputs
*/
    ssSetNumOutputs(       S, 113);  /* number of outputs
*/
    ssSetDirectFeedThrough(S, 1);   /* direct feedthrough flag
*/
    ssSetNumSampleTimes(   S, 1);   /* number of sample times
*/
    ssSetNumSFcnParams(    S, 5);   /* number of input arguments
*/
    ssSetNumRWork(         S, 0);   /* number of real work vector
elements   */
    ssSetNumIWork(         S, 0);   /* number of integer work vector
elements*/
    ssSetNumPWork(         S, 0);   /* number of pointer work vector
elements*/
}

/*
 * mdlInitializeSampleTimes - initialize the sample times array
 */
static void mdlInitializeSampleTimes(SimStruct *S)
{
    ssSetSampleTime(S, 0, CONTINUOUS_SAMPLE_TIME);
    ssSetOffsetTime(S, 0, 0.0);
}


/*
```

```
 * mdlInitializeConditions - initialize the states
 */
static void mdlInitializeConditions(double *x0, SimStruct *S)
{
int i;

for (i = 0; i < 80; i++) {
   x0[i] = mxGetPr(XINIT)[i];
}

}

/*
 * mdlOutputs - compute the outputs
 */

static void mdlOutputs(double *y, double *x, double *u, SimStruct
*S, int tid)
{
  double gamma, gamma_eff, modeltype;
  int i;

  gamma = x[9]/u[13];
  gamma_eff = x[0]/u[13];

  modeltype = mxGetPr(MODELTYPE)[0];

  if (modeltype < 0.5) {
     /* underflow */
     y[0]=x[19];
     y[1]=x[29];
     y[2]=u[2]*gamma;
     y[3]=u[3]*gamma;
     y[4]=u[4]*gamma;
     y[5]=u[5]*gamma;
     y[6]=u[6]*gamma;
     y[7]=x[39]; /* use oxygen in return sludge flow */
     y[8]=x[49];
     y[9]=x[59];
     y[10]=x[69];
     y[11]=u[11]*gamma;
     y[12]=x[79];
     y[13]=x[9];
     y[14]=u[15];  /* Q_r */
     y[15]=u[16];  /* Q_w */

     /* effluent */
     y[16]=x[10];
     y[17]=x[20];
     y[18]=u[2]*gamma_eff;
     y[19]=u[3]*gamma_eff;
     y[20]=u[4]*gamma_eff;
     y[21]=u[5]*gamma_eff;
     y[22]=u[6]*gamma_eff;
     y[23]=x[30]; /* use oxygen in effluent flow */
     y[24]=x[40];
     y[25]=x[50];
     y[26]=x[60];
     y[27]=u[11]*gamma_eff;
     y[28]=x[70];
```

```
   y[29]=x[0];
   y[30]=u[14]-u[15]-u[16];  /* Q_e */

   /* internal TSS states */
   y[31]=x[0];
   y[32]=x[1];
   y[33]=x[2];
   y[34]=x[3];
   y[35]=x[4];
   y[36]=x[5];
   y[37]=x[6];
   y[38]=x[7];
   y[39]=x[8];
   y[40]=x[9];

   y[41]=gamma;
   y[42]=gamma_eff;

   for (i = 10; i < 80; i++)
      y[i+33] = x[i];
}

else if ((modeltype > 0.5) && (modeltype < 1.5)) {
   /* underflow */
   y[0]=x[10];
   y[1]=x[20];
   y[2]=u[2]*gamma;
   y[3]=u[3]*gamma;
   y[4]=u[4]*gamma;
   y[5]=u[5]*gamma;
   y[6]=u[6]*gamma;
   y[7]=x[30]; /* use oxygen in return sludge flow */
   y[8]=x[40];
   y[9]=x[50];
   y[10]=x[60];
   y[11]=u[11]*gamma;
   y[12]=x[70];
   y[13]=x[9];
   y[14]=u[15];  /* Q_r */
   y[15]=u[16];  /* Q_w */

   /* effluent */
   y[16]=x[10];
   y[17]=x[20];
   y[18]=u[2]*gamma_eff;
   y[19]=u[3]*gamma_eff;
   y[20]=u[4]*gamma_eff;
   y[21]=u[5]*gamma_eff;
   y[22]=u[6]*gamma_eff;
   y[23]=x[30]; /* use oxygen in effluent flow */
   y[24]=x[40];
   y[25]=x[50];
   y[26]=x[60];
   y[27]=u[11]*gamma_eff;
   y[28]=x[70];
   y[29]=x[0];
   y[30]=u[14]-u[15]-u[16];  /* Q_e */

   /* internal TSS states */
   y[31]=x[0];
```

```
   y[32]=x[1];
   y[33]=x[2];
   y[34]=x[3];
   y[35]=x[4];
   y[36]=x[5];
   y[37]=x[6];
   y[38]=x[7];
   y[39]=x[8];
   y[40]=x[9];

   y[41]=gamma;
   y[42]=gamma_eff;

   for (i = 10; i < 20; i++)
      y[i+33] = x[10];
   for (i = 20; i < 30; i++)
      y[i+33] = x[20];
   for (i = 30; i < 40; i++)
      y[i+33] = x[30];
   for (i = 40; i < 50; i++)
      y[i+33] = x[40];
   for (i = 50; i < 60; i++)
      y[i+33] = x[50];
   for (i = 60; i < 70; i++)
      y[i+33] = x[60];
   for (i = 70; i < 80; i++)
      y[i+33] = x[70];
}

else if (modeltype > 1.5) {
   /* underflow */
   y[0]=u[0];
   y[1]=u[1];
   y[2]=u[2]*gamma;
   y[3]=u[3]*gamma;
   y[4]=u[4]*gamma;
   y[5]=u[5]*gamma;
   y[6]=u[6]*gamma;
   y[7]=u[7]; /* use oxygen in return sludge flow */
   y[8]=u[8];
   y[9]=u[9];
   y[10]=u[10];
   y[11]=u[11]*gamma;
   y[12]=u[12];
   y[13]=x[9];
   y[14]=u[15];  /* Q_r */
   y[15]=u[16];  /* Q_w */

   /* effluent */
   y[16]=u[0];
   y[17]=u[1];
   y[18]=u[2]*gamma_eff;
   y[19]=u[3]*gamma_eff;
   y[20]=u[4]*gamma_eff;
   y[21]=u[5]*gamma_eff;
   y[22]=u[6]*gamma_eff;
   y[23]=u[7]; /* use oxygen in effluent flow */
   y[24]=u[8];
   y[25]=u[9];
   y[26]=u[10];
```

```
      y[27]=u[11]*gamma_eff;
      y[28]=u[12];
      y[29]=x[0];
      y[30]=u[14]-u[15]-u[16];  /* Q_e */

      /* internal TSS states */
      y[31]=x[0];
      y[32]=x[1];
      y[33]=x[2];
      y[34]=x[3];
      y[35]=x[4];
      y[36]=x[5];
      y[37]=x[6];
      y[38]=x[7];
      y[39]=x[8];
      y[40]=x[9];

      y[41]=gamma;
      y[42]=gamma_eff;

      for (i = 10; i < 20; i++)
         y[i+33] = u[0];
      for (i = 20; i < 30; i++)
         y[i+33] = u[1];
      for (i = 30; i < 40; i++)
         y[i+33] = u[7];
      for (i = 40; i < 50; i++)
         y[i+33] = u[8];
      for (i = 50; i < 60; i++)
         y[i+33] = u[9];
      for (i = 60; i < 70; i++)
         y[i+33] = u[10];
      for (i = 70; i < 80; i++)
         y[i+33] = u[12];
   }
}

/*
 * mdlUpdate - perform action at major integration time step
 */

static void mdlUpdate(double *x, double *u, SimStruct *S, int tid)
{
}

/*
 * mdlDerivatives - compute the derivatives
 */
static void mdlDerivatives(double *dx, double *x, double *u,
SimStruct *S, int tid)
{

double v0_max, v0, r_h, r_p, f_ns, X_t, area, h, feedlayer, volume,
modeltype;
double Q_f, Q_e, Q_u, v_up, v_dn, v_in, eps;
int i;
double vs[10];
double Js[11];
double Jstemp[10];
double Jflow[11];
```

```
v0_max = mxGetPr(PAR)[0];
v0 = mxGetPr(PAR)[1];
r_h = mxGetPr(PAR)[2];
r_p = mxGetPr(PAR)[3];
f_ns = mxGetPr(PAR)[4];
X_t = mxGetPr(PAR)[5];
area = mxGetPr(DIM)[0];
h = mxGetPr(DIM)[1]/mxGetPr(LAYER)[1];
feedlayer = mxGetPr(LAYER)[0];
modeltype = mxGetPr(MODELTYPE)[0];
volume = area*mxGetPr(DIM)[1];

eps = 0.01;
v_in = u[14]/area;
Q_f = u[14];
Q_u = u[15] + u[16];
Q_e = u[14] - Q_u;
v_up = Q_e/area;
v_dn = Q_u/area;

for (i = 0; i < 10; i++) {
   vs[i] = v0*(exp(-r_h*(x[i]-f_ns*u[13]))-exp(-r_p*(x[i]-
f_ns*u[13])));
   if (vs[i] > v0_max)
      vs[i] = v0_max;
   else if (vs[i] < 0)
      vs[i] = 0;
}

for (i = 0; i < 10; i++) {
   Jstemp[i] = vs[i]*x[i];
}

for (i = 0; i < 11; i++) {
   if (i < (feedlayer-eps))
      Jflow[i] = v_up*x[i];
   else
      Jflow[i] = v_dn*x[i-1];
}

Js[0] = 0;
Js[10] = 0;
for (i = 0; i < 9; i++) {
   if ((i < (feedlayer-1-eps)) && (x[i+1] <= X_t))
      Js[i+1] = Jstemp[i];
   else if (Jstemp[i] < Jstemp[i+1])
      Js[i+1] = Jstemp[i];
   else
      Js[i+1] = Jstemp[i+1];
}

for (i = 0; i < 10; i++) {
   if (i < (feedlayer-1-eps))
      dx[i] = (-Jflow[i]+Jflow[i+1]+Js[i]-Js[i+1])/h;
   else if (i > (feedlayer-eps))
      dx[i] = (Jflow[i]-Jflow[i+1]+Js[i]-Js[i+1])/h;
   else
      dx[i] = (v_in*u[13]-Jflow[i]-Jflow[i+1]+Js[i]-Js[i+1])/h;
}
```

```
/* soluble component S_I */
if (modeltype < 0.5) {
   for (i = 10; i < 20; i++) {
      if (i < (feedlayer-1+10-eps))
         dx[i] = (-v_up*x[i]+v_up*x[i+1])/h;
      else if (i > (feedlayer+10-eps))
         dx[i] = (v_dn*x[i-1]-v_dn*x[i])/h;
      else
         dx[i] = (v_in*u[0]-v_up*x[i]-v_dn*x[i])/h;
   }
}
else if ((modeltype > 0.5) && (modeltype < 1.5)) {
   dx[10] = (Q_f*(u[0]-x[10]))/volume;
   for (i = 11; i < 20; i++)
      dx[i] = 0;
}
else if (modeltype > 1.5) {
   for (i = 10; i < 20; i++)
      dx[i] = 0;
}


/* soluble component S_S */
if (modeltype < 0.5) {
   for (i = 20; i < 30; i++) {
      if (i < (feedlayer-1+20-eps))
         dx[i] = (-v_up*x[i]+v_up*x[i+1])/h;
      else if (i > (feedlayer+20-eps))
         dx[i] = (v_dn*x[i-1]-v_dn*x[i])/h;
      else
         dx[i] = (v_in*u[1]-v_up*x[i]-v_dn*x[i])/h;
   }
}
else if ((modeltype > 0.5) && (modeltype < 1.5)) {
   dx[20] = (Q_f*(u[1]-x[20]))/volume;
   for (i = 21; i < 30; i++)
      dx[i] = 0;
}
else if (modeltype > 1.5) {
   for (i = 20; i < 30; i++)
      dx[i] = 0;
}

/* soluble component S_O */
if (modeltype < 0.5) {
   for (i = 30; i < 40; i++) {
      if (i < (feedlayer-1+30-eps))
         dx[i] = (-v_up*x[i]+v_up*x[i+1])/h;
      else if (i > (feedlayer+30-eps))
         dx[i] = (v_dn*x[i-1]-v_dn*x[i])/h;
      else
         dx[i] = (v_in*u[7]-v_up*x[i]-v_dn*x[i])/h;
   }
}
else if ((modeltype > 0.5) && (modeltype < 1.5)) {
   dx[30] = (Q_f*(u[7]-x[30]))/volume;
   for (i = 31; i < 40; i++)
      dx[i] = 0;
}
else if (modeltype > 1.5) {
   for (i = 30; i < 40; i++)
```

```
      dx[i] = 0;
}

/* soluble component S_NO */
if (modeltype < 0.5) {
   for (i = 40; i < 50; i++) {
      if (i < (feedlayer-1+40-eps))
         dx[i] = (-v_up*x[i]+v_up*x[i+1])/h;
      else if (i > (feedlayer+40-eps))
         dx[i] = (v_dn*x[i-1]-v_dn*x[i])/h;
      else
         dx[i] = (v_in*u[8]-v_up*x[i]-v_dn*x[i])/h;
   }
}
else if ((modeltype > 0.5) && (modeltype < 1.5)) {
   dx[40] = (Q_f*(u[8]-x[40]))/volume;
   for (i = 41; i < 50; i++)
      dx[i] = 0;
}
else if (modeltype > 1.5) {
   for (i = 40; i < 50; i++)
      dx[i] = 0;
}

/* soluble component S_NH */
if (modeltype < 0.5) {
   for (i = 50; i < 60; i++) {
      if (i < (feedlayer-1+50-eps))
         dx[i] = (-v_up*x[i]+v_up*x[i+1])/h;
      else if (i > (feedlayer+50-eps))
         dx[i] = (v_dn*x[i-1]-v_dn*x[i])/h;
      else
         dx[i] = (v_in*u[9]-v_up*x[i]-v_dn*x[i])/h;
   }
}
else if ((modeltype > 0.5) && (modeltype < 1.5)) {
   dx[50] = (Q_f*(u[9]-x[50]))/volume;
   for (i = 51; i < 60; i++)
      dx[i] = 0;
}
else if (modeltype > 1.5) {
   for (i = 50; i < 60; i++)
      dx[i] = 0;
}

/* soluble component S_ND */
if (modeltype < 0.5) {
   for (i = 60; i < 70; i++) {
      if (i < (feedlayer-1+60-eps))
         dx[i] = (-v_up*x[i]+v_up*x[i+1])/h;
      else if (i > (feedlayer+60-eps))
         dx[i] = (v_dn*x[i-1]-v_dn*x[i])/h;
      else
         dx[i] = (v_in*u[10]-v_up*x[i]-v_dn*x[i])/h;
   }
}
else if ((modeltype > 0.5) && (modeltype < 1.5)) {
   dx[60] = (Q_f*(u[10]-x[60]))/volume;
   for (i = 61; i < 70; i++)
      dx[i] = 0;
}
```

```
else if (modeltype > 1.5) {
   for (i = 60; i < 70; i++)
      dx[i] = 0;
}

/* soluble component S_ALK */
if (modeltype < 0.5) {
   for (i = 70; i < 80; i++) {
      if (i < (feedlayer-1+70-eps))
         dx[i] = (-v_up*x[i]+v_up*x[i+1])/h;
      else if (i > (feedlayer+70-eps))
         dx[i] = (v_dn*x[i-1]-v_dn*x[i])/h;
      else
         dx[i] = (v_in*u[12]-v_up*x[i]-v_dn*x[i])/h;
   }
}
else if ((modeltype > 0.5) && (modeltype < 1.5)) {
   dx[70] = (Q_f*(u[12]-x[70]))/volume;
   for (i = 71; i < 80; i++)
      dx[i] = 0;
}
else if (modeltype > 1.5) {
   for (i = 70; i < 80; i++)
      dx[i] = 0;
}

}



/*
 * mdlTerminate - called when the simulation is terminated.
 */
static void mdlTerminate(SimStruct *S)
{
}

#ifdef  MATLAB_MEX_FILE    /* Is this file being compiled as a MEX-
file? */
#include "simulink.c"      /* MEX-file interface mechanism */
#else
#include "cg_sfun.h"       /* Code generation registration function
*/
#endif
```

Appendix D – Simulink model of DO control